In Search of Excellence: SHOA as a Competitive Shrike Optimization Algorithm for Multimodal Problems

Hanan K. AbdulKarim[1], Tarik A. Rashid[2]
[1]Software Engineering Department, Engineering College, University of Salahaddin - Erbil, Erbil, Iraq
[2]Computer Science and Engineering Department, University of Kurdistan Hewler, Erbil, Iraq

Corresponding author: Hanan K. AbdulKarim (email: hanan.abdulkarim@su.edu.krd)

ABSTRACT In this paper, a swarm intelligence optimization algorithm is proposed as the Shrike Optimization Algorithm (SHOA). Many creatures living in a group and surviving for the next generation randomly search for foods; they follow the best one in the swarm, called swarm intelligence. Swarm-based algorithms are designed to mimic creatures' behaviors, but in the multi-modal problem competition, they lack the ability to find optimal solutions in many cases. The main inspiration for the proposed algorithm is taken from the swarming behavior of shrike birds in nature. The shrike birds are migrating from their territory to survive. However, the SHOA mimics the surviving behavior of shrike birds for living, adaptation, and increasing. Two parts of optimization exploration and exploitation are designed by modeling shrike breeding and searching for foods to feed nestlings until they get ready to fly and live independently. This paper is a mathematical model for the SHOA to perform optimization. The SHOA benchmarked 29 competitive, well-known mathematical test functions and four real-world engineering problems with different conditions, both constrained and unconstrained. The statistical results show the proposed SHOA can perform nearly ideal results when compared with other well-known algorithms for multi-modal problems. The results for engineering optimization problems show the SHOA outperforms other nature-inspired algorithms.

INDEX TERMS Shrike, Optimization, Constrained Optimization, swarm Intelligence, multi-modal, meta-heuristic, population-based optimization, engineering problem.

I. INTRODUCTION

Optimization techniques have become important in the last few decades. Optimization is finding the best optimal or semi-optimal solution by achieving a specific objective without violating constraints. In some cases, no objective functions exist, but a feasible solution depending on constraints is an optimal solution, called a feasibility problem. Many complex and rough-solvable problems in engineering, science, medicine, statistics, and computer science have been solved by optimization algorithms within a short time. Mathematical calculation and programs have been used to solve such a problem, but recently, for solving complex problems, some meta-heuristic optimization algorithms have been used to find acceptable solutions. Many optimization algorithms are nature-inspired algorithms designed by mimicking creatures from nature; many of those algorithms depend on swarms' social behavior and are called swarm-based algorithms.

Optimization algorithms have been classified as single-based and population-based. Single-based optimization searches for an optimum solution using a single solution like simulated Annealing (SA), Hill Climbing (HC), Variable Neighborhood Search (VNS), and Tabu Search (TS) [1–4], while the population-based optimization algorithms using a group of solutions as population and searching around number of the neighbor of the solutions in the search space, but is should has good exploration and exploitation techniques to not trap in the local optima, population-based like Genetic Algorithm (GA) [5], Differential Evaluation (DE) [6], Genetic Programming (GP) [7], and swarm-based algorithms.

Swarm-based algorithms are stochastic because they work on the Swarm Intelligence (SI) of the creature's behavior. Ant Colony Optimization (ACO) [8] is an old algorithm that studies the collective behavior of ants searching for food sources. It simply translates the fact that every ant has its own decision for foraging on a specific path, each ant signs the path by pheromone when it transits to the food source, and it will add pheromone again when returning to the nest, so other ants will take a path with higher pheromone and leave their path, then the shortest path will be accomplished by leaving the low pheromone path and use the higher-level pheromones path. Ant System (AS) [9,10] applied to solve various combinatorial optimization problems. The application of AS includes the Traveling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP), and the Job-shop

Scheduling Problem (JSP), it shows the ability to solve those problems, also applied in the classification field [11], and cloud computing [12].

Particle Swarm Optimization (PSO) [13] mimics the inspiration of SI of birds, and fish while the author considered birds, simple techniques were used that birds follow the flock fly direction, the best food source obtained so far, and the best food sources that swarm found, simply it uses rules to find the best solution in the search space, and it is applied in many fields of design, image processing, and others [14,15] which successfully improve solutions. The society and Civilization algorithm [16] is the adaption of societies simulated for optimization problem-solving. Artificial Bee Colony (ABC) works on honey bees finding food sources in [17,18], it works on how explored bees find food sources and share information with employed bees, the onlooker bees exploit food sources more to find better sources and keep the best food source, ABC outperforms many optimization algorithms for some optimization problems of global optimization, feature selection, neural network fields, vehicle routing [19–23]. Bacterial Foraging Behavior (BFO) the bacterial foraging behavior has been a source for development, applied for electrical power filter problems, and designed fuzzy control for the system [24–26]. Firefly Algorithm (FA), the flashing light and attractiveness of fireflies were formulated as FA algorithm, used to solve multi-modal problems, design structure, and many other applications [27,28]. The moth-flame Optimization (MFO) Algorithm [29] was developed by studying moths' navigation in nature and how they move around lights. Solving problems with clustering suffers from exploration the MFO is added to handle the clustering problem[30]. Since the reviewed methods initially were proposed, the researchers have worked to enhance or implement them in many domains and for various challenges [31–39]. Recently, many researchers working to improve old and new algorithms to apply to variant types of problems.

The effective application of swarm-based algorithms by the scientific and business communities has demonstrated the worth of these methods in practice. The benefits of SI-based algorithms are the reasoning success of the algorithms mentioned before. Swarm-based optimization methods work with groups as a population and have some randomness during searching for a solution. A population has some drawbacks as it needs more computation time, but nowadays, high-speed processors and parallel programming will solve this problem. Despite all optimization algorithms, there is no universal algorithm used to solve all optimization problems. Still, some algorithms outperform others in many types of optimization problems. The researchers are working to find an algorithm that outperforms other algorithms for most of the problems or find new algorithms that can solve unsolved problems.

This paper proposes a novel swarm-based SHOA to increase the number of solved multi-modal and complex problems because none of the optimization algorithms can solve all problems. Depending on the nature of the problem, a specific algorithm must be applied to find the best solution. Intensification and diversification are the essential components of meta-heuristic algorithms. The main contributions of this study are:

1. The proposed SHOA is designed for multi-modal problems by finding many local optima and keeping them to produce global optima because multi-modal problems have many local optima and many optimization algorithms lack the ability to find the global optima.
2. In the SHOA mathematical proposal produced, depending on the shrike bird’s physical simulations of a parent bird's dominance in a specific life stage, the roles of female and male birds were separated depending on reality and life style.
3. The SHOA, applying randomization will diverge the algorithm from the current solution, which is considered a local optimum, and redirect the algorithm to search the space globally to increase diversity, while finding a solution during the local search by choosing the best solution so far will converge the algorithm to an optimal solution, increasing convergence.

The remainder of this article is structured as follows: (Part II) presents inspiration from shrike birds and a mathematical model for the proposed SHOA (Part III) results and discussion on comparative benchmarks and some competitive functions, and (Part IV) SHOA applied real-world cases, studies as engineering problems and the performance compared with other optimization algorithms, finally (Part V) conclude the work of this study and show direction for coming studies.

II. SHRIKE BIRDS

A. SOURCES OF INSPIRATION FOR SHRIKES

The Laniidae family of passerine birds includes shrikes, which are distinguished by their propensity to impale their flesh on thorns after capturing insects, small birds, or animals. The shrikes are two genera with 34 species distributed throughout the world. In North America, there is a member of the Shrike family called Loggerhead Shrikes. Loggerhead shrikes, also called butcherbirds and migrating shrikes, reach a weight of roughly 48 grams [40–42]. Within the Laniidae family, this remarkable bird is rather huge, and its large head may have contributed to its unique name. Males and females have similar appearances; it is difficult to distinguish between them. They have black, white, and gray markings on their bodies and a black mask that covers their eyes [43].

Over its range, the loggerhead's appearance varies slightly by region. Loggerheads eat mainly small vertebrates and small mammals. They live, migrate, eat in population, and use cooperative breeding [41]. Make nests on the trees; the female will deposit between four and seven eggs in a clutch, which she will then incubate for roughly sixteen days [43]. For a period of seventeen to twenty days, both parents are responsible for taking care of the nestlings. After leaving the

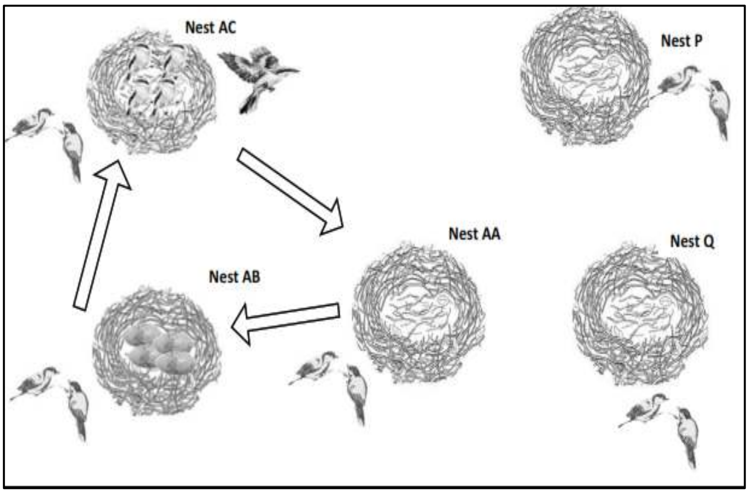

Figure 1 Shrike bird life cycle

nest, the young birds remain close to their parents for three weeks, during which time they get food from both parents, develop their flight, and at night return to be warmed by the parents. For more information, return to reference [40]. The population of the shrike bird life cycle is simulated in Figure(1). There are three nests: A, P, and Q are the population of birds' nests; the nest AA parent will brood eggs at the nest (AB); the nestling will grow up and become adults ready to fly and later depend on themselves, the breeding and surviving of the birds of nest A shown from (A to C).

B. SHRIKE OPTIMIZATION ALGORITHM

Depending on the nesting and reproductive behavior of the shrike birds explained in the previous section, the shrikes live in a population out of the urban area; the population has many nests, and each nest starts with two birds as parents. The breeding and surviving behaviors of the shrikes were mimicked by Shrike Optimization (SHO) algorithm.

The SHO algorithms start by initializing parameters, where n is the size of nests, m is the number of eggs considered nestlings surviving, and α is constant considered a natural factor affecting the bird. The search space of SHOA starts with a population of n nests, where each nest starts with two parent birds generated randomly using the pseudo-code shown in the algorithm (2).
After the population is generated and nests are ready, seven nestlings will be generated by the pseudo-code generated nestling shown in the algorithm (3). The nestlings will depend on their parents; the male parent is dominant, which feeds the female, and for the nestlings to survive, it will feed by itself; and female also feeds by itself, feeding the nestling if they didn't get food from the male parent. In this algorithm, for every bird (parent, nestling), calculate αmax using equation (1), and generate a random number r. These parameters are used as natural outside factors in feeding.

$$\alpha_{max} = \begin{cases} x - LB & if\ (x - LB) > (UB - x) \\ UB - x & else \end{cases} \quad (1)$$

Where x is used as a solution, either parent or nestling, LB is the lower bound of the search space and UB is the upper bound, α_{max} is the maximize factor, and r simple random value in the range [0,1].

For all birds, if they are parents the Δfood is generated by formula (2), which multiplies the current state by a random factor.

$$\Delta food_j = bird_j * r \quad (2)$$

But for feeding nestlings and female birds, the Δfood is generated by formula (3), which is the current bird state with a male parent bringing food and using α_{max}, r factors.

$$\Delta \mathrm{food}_j = \mathrm{r} * (bird_j - M_{parent} + \alpha_{max}) \quad (3)$$

Whereas the nestlings didn't survive by food from the male parent, then they tried to survive through the female parent using formula (4), the same as formula (3), but cos (α) was used as a constant factor of nature.

$$\Delta \text{food}_j = r * (bird_j - F_{parent} + cos(\alpha)) \qquad (4)$$

After generating food, the birds' next status will be calculated using formula (5), which is the current state of birds getting food.

$$\text{bird}_j{}^{t+1} = \text{bird}_j{}^{t} + \Delta \text{food}_j \qquad (5)$$

Calculate fitness for the birds, if better than the current, the current state will update with the new state, this is done because, in reality, not all birds get food at the same time.

The SHO algorithm keeps the best of each nest as the local best, then the population keeps the best of all local best as global. The idea of multi-modality is used here; there will be many local bests and one global best. Every three to five generations, the algorithm will keep just two birds as parents and remove all other birds as they die or fly far away from nests as they get ready to live independently. Every female parent starts laying eggs using formulas (6) and (7), getting food, and random $r \in [0,1]$ factor.

$$\Delta \text{food}_j = (F_{parent} - M_{parent}) + r \qquad (6)$$

$$egg_j = F_{parent} + \Delta \text{food}_j \qquad (7)$$

Generation after generation of searches are conducted using a randomly generated population. The best populations share knowledge to produce the next generation, which is the most robust characteristic of biology-based algorithms used in the proposed SHOA. Future generations will be able to come up with better solutions due to this reality. The flow chart also specifies the process of the SHOA and the overall steps are presented in Appendix B Figures (6,7) as flowchart of SHOA.

III. IMPLEMENTATION AND RESULTS

In this section, the proposed SHOA is benchmarked on the number of global optimization test functions to show the performance of SHOA and the results compared with some recently developed optimization algorithms. Four groups of test functions are selected as uni-modal, multi-modal, complex, and 100-digit Challenge test functions [44–49], each having a specific characteristic. The test functions were shifted and rotated by the values shown in tables (1-3) to increase the complexity of the problems, and to show the performance of the proposed algorithm. The results compared with MFO and Fitness Dependent Optimizer (FDO) [29,50,51], which are the most recent and interesting algorithms because of their performance over other algorithms. MFO are recently more interested algorithm, it has fast convergence but it has exploitation problems that occur with complex and composite problems, the reason for the high exploration of the MFO algorithm [39].

Despite the increasing complexity of the tested functions by rotating and shifting, all uni-modal test functions have a single optimum solution, while increasing the dimension will increase the problem difficulty and computation time to reach global optimum solution. The test functions F1-F7 shown in Appendix Table 12 uni-modal function are considered in this study. The multi-modal function has many local optima, which increases the difficulty of the algorithm to find an optimum solution because of trapping in local optima. The test functions F8-13 in the Appendix Table 13 multi-modal test functions are considered multi-modal problems for comparison, and the specification of functions, rotation, and shift values of the problems are specified in the multi-modal table.

Composition functions are compounds of many functions with rotation, shifting, and add function bias. These functions are important as case studies because the properties of multi-functions are mixed like real-world problems, and they will show the performance of the algorithm in the exploration and exploitation search capability. The test functions F14-F19 mentioned in Table 14 composite functions had f_{min}=0 shown, where σ is used to coverage range control of each f(x), and λ used for compress and stretch the function, all these are tested with composite functions in the current study.

Recently, many competition functions have been provided by high-impact conferences to be used as comparison studies for competing for the performance of optimization algorithms. The 100-Digit functions challenge has 10 hard-solved problems as compound functions from the Society for Industrial and Applied Mathematics (SIAM), the purpose of solving such problem in this paper is to find the optimum solution within a specific time because, in the original paper, there is no time limitation for solve problems [49]. The problems are shown in Table 15 the Hundred-Digit Challenge basics with the range of x values, and dimensions of the test problems.

Numerical examples with dimensions specified in the tables, each bunch mark run 30 times with SHOA. The SHOA runs with a population size are 15, each nest starts with two shrikes as a parent of the nest, parents breeding 7 eggs, nestling birds feeding by the parent by 4 generations of algorithm cycles, then best 2 keep for next generation, other birds removed from nest, 500 iterations specified for each turn the SHOA execution. The statical results Mean is the mean value over 30 turns, and Std is the standard deviation summarized in Tables (1,2,3, and 4).

Table 1 Uni-modal Comparison Results

F	SHOA		MFO		FDO	
	Mean	STD	Mean	STD	Mean	STD
F1	5.03E-11	7.57E-11	1.17E-04	1.50E-04	**7.47E-21**	7.26E-19
F2	1.52E-05	1.67E-05	6.39E-04	8.77E-04	**9.39E-06**	6.91E-06
F3	**2.23E-09**	4.27E-09	6.97E+02	1.89E+02	8.55E-07	4.40E-06
F4	**1.94E-07**	1.45E-07	7.07E+01	5.28E+00	6.69E-04	2.49E-03
F5	**3.16E-09**	6.20E-09	1.39E+02	1.20E+02	2.35E+01	5.98E+01
F6	**0.00E+00**	0	1.13E-04	9.87E-05	1.42E-18	4.75E-18
F7	5.03E-1	7.57E-11	**9.12E-02**	4.64E-02	5.44E-01	3.15E-01

The proposed algorithm is designed for multi-modal, but uni-modal test functions are also benchmarked to show the performance of SHOA, in the comparison F1, F2, and F7 are quite good compared with other algorithms, from F3-F6 the SHOA outperforms other algorithms, the comparative results shown statistically the Table 1.

In Table 2 as a multi-modal comparative, SHOA shows superiority over another algorithm in all functions just F10, but in the F10 the fitness is near to optimal by 10-8 is also a good performance. While comparing results with the composite functions shown in Table 3, novel SHOA outperforms other powerful algorithms in all F15-F19, but in F14 it is quite good. In the all-test function, SHOA performs a good solution outperforming many algorithms, even in F1, F2, F10, and F14 the function fitness is ≤×10-5 which means too close to an optimal value.

Table 2 Multi-modal Comparison Results with Algorithms

F	SHOA		MFO		FDO	
	Mean	STD	Mean	STD	Mean	STD
F8	**-3.1E+06**	1.4E-08	-8.50E+03	7.2E+02	-2.29E+06	2.0E+05
F9	**2.88E-08**	3.2E-08	8.46E+01	1.6E+01	1.46E+01	5.2E+00
F10	3.00E-08	2.3E-08	1.26E+00	7.3E-01	**4.00E-15**	6.3E-16
F11	**4.27E-10**	1.8E-09	1.91E-02	2.1E-02	5.69E-01	1.0E-01
F112	**1.23E-10**	2.4E-10	8.94E-01	8.8E-01	1.98E+01	2.6E+01
F13	**4.63E-11**	5.2E-11	1.16E-01	1.9E-01	1.03E+01	7.4E+00

Table 3 Composition Modal Comparison Results

F	SHOA		MFO		FDO	
	Mean	STD	Mean	STD	Mean	STD
F14	5.62E-14	1.1E-13	**8.25E-31**	1.0E-30	3.79E-07	6.3E-07
F15	**9.53E-15**	2.5E-14	6.67E+01	5.3E+01	1.50E-03	1.2E-03
F16	**1.94E-12**	2.8E-12	1.19E+02	2.8E+01	6.38E-03	1.0E-02
F17	**1.18E-03**	1.9E-03	3.45E+02	4.3E+01	2.38E+01	2.1E-01
F18	**3.06E-04**	3.1E-04	1.04E+01	3.7E+00	2.23E+02	9.9E-06
F19	**9.29E-04**	1.1E-03	7.07E+02	1.9E+02	2.28E+01	1.0E-02

Table 4100-Digit Challenge Problem Comparison Results

F	SHOA		FDO		AZOA	
	Mean	STD	Mean	STD	Mean	STD
C01	**9.42E+02**	3.6E+02	4.59E+03	2.0E+04	8.16E+04	1.5E+05
C02	**3.00E+00**	1.3E-06	4.00E+00	3.2E-09	1.73E+01	3.4E−05
C03	**3.35E+00**	1.2E+00	1.37E+01	1.6E-11	1.27E+01	**2.8E−06**
C04	1.14E+02	1.4E+01	**3.41E+01**	1.6E+01	2.67E+02	1.9E+02
C05	**6.17E-01**	9.4E-02	2.14E+00	8.5E-02	1.42E+00	2.5E−01
C06	**7.92E+00**	9.1E-01	1.21E+01	6.0E-01	9.67E+00	1.1E+00
C07	1.64E+02	1.3E+01	**1.20E+02**	1.3E+01	2.45E+02	2.0E+02
C08	**3.07E+00**	1.7E-01	6.10E+00	7.5E-01	5.06E+00	6.1E−01
C09	2.51E+00	5.1E-01	**2.00E+00**	1.5E-10	3.32E+00	4.9E−01
C10	**1.72E+00**	8.8E-16	2.72E+00	8.8E-16	1.97E+01	2.6E+00

In Table 4 for Hundred-Digit challenge problems, it can be seen the SHOA outperforms other optimization algorithms in seven CEC 2019 functions. Indeed, results in many test functions like (CE19-01, CE19-05, and CE19-06) show the novel SHOA is more powerful than other algorithms not only at the average value of 30 runs but at other statistical Std values. Once again, the signed rank test (as shown in Tables 5-6) demonstrated the superior performance of SHOA in solving all 29 test problem functions.

Furthermore, Figure 2 optimal values generated display variation with an optimal value generated by algorithms, showing the MFO reach 700, FDO > 200, while SHOA is close to zero with all 19 functions of 3 groups as uni-modal, multi-modal, and Composite test functions.

Table 5 Ranking Value of Test Problem Function

Functions	SHOA Rank	MFO Rank	FDO Rank
F1	2	3	1
F2	2	3	1
F3	1	3	2
F4	1	3	2
F5	1	3	2
F6	1	3	2
F7	2	1	3
F8	1	3	2
F9	1	3	2
F10	2	3	1
F11	1	2	3
F112	1	2	3
F13	1	2	3
F14	2	1	3
F15	1	3	2
F16	1	3	2
F17	1	3	2
F18	1	2	3
F19	1	3	2
total	**24**	49	41
total rank/ no. of fun	24/19	49/19	41/19
Ranking	**1.26**	2.58	2.16

Table 6 Comparison Assigned Ranking of CEC-19 Test Problems

function	SHOA Rank	MFO Rank	AZOA Rank
C01	1	2	3
C02	1	2	3
C03	1	3	2
C04	2	1	3
C05	1	3	2
C06	1	2	3
C07	2	1	3
C08	1	3	2
C09	2	1	3
C10	1	3	2
Total	13	21	26
total rank/ no. of fun	13/10	21/10	26/10
Ranking	**1.3**	2.1	2.6

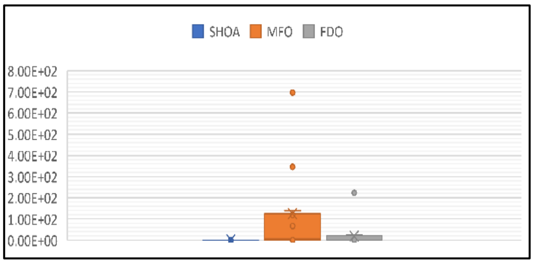

Figure 2 Optimal value

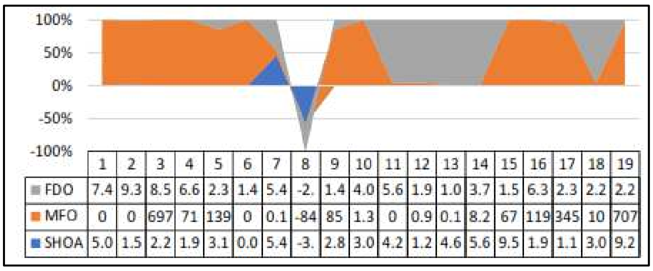

Figure 3 Percentage Contribution Area All functions

In Figure 3 the contribution area clearly shows that SHOA occupies less percentage of the total area because it minimizes problems, then a large area means less performance.

The 100-digit problems percent contribution area is shown in Figure 4 which the CE19_04 and CE19_07 take more area than FDO while in all other cases, SHOA has less occupation area. The performance and trend line of some functions are shown in Figure 5.

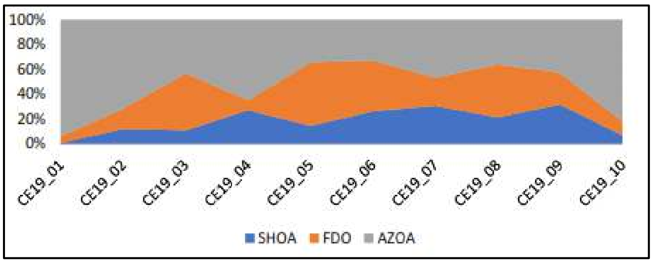

Figure 4 Percentage Contribution Area CE19 Functions

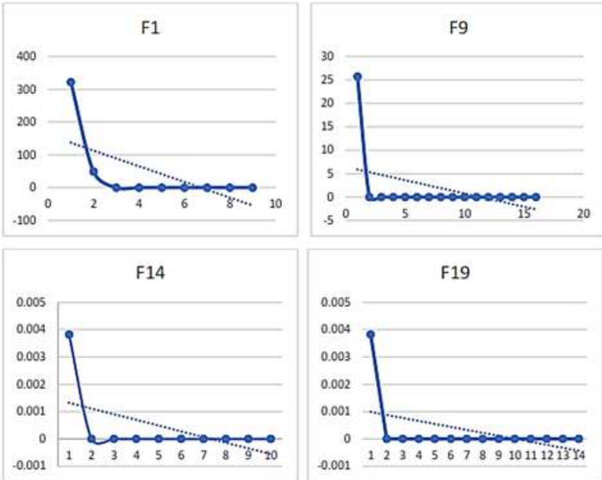

Figure 5 Sample of Functions performance and trend line

IV. ENGINEERING PROBLEMS SOLVING

In this study, four constrained engineering problems, namely three-bar truss design, gear train design, antenna array design, and frequency-modulated sound wave design, are considered to investigate the applicability of SHOA.

The problems have equality and inequality constraints, the SHOA should be equipped with the constrained solutions. Although, in constraint problem solving there will be feasible and infeasible solutions, to investigate infeasible solutions, some algorithms use penalty functions [52], in this study the

death penalty is used, and the infeasible solutions are discarded and not investigated with a penalty to speed up the algorithm process. It is worth noting that the population size is set to 15, and iterations set to 500, for 30 turns for all the problems in this section.

A. GEAR TRAIN DESIGN PROBLEM SOLVING

The gear train design is a mechanical engineering problem, the main objective is to minimize the desired ratio with the current ratio [53], the objective function was formulated as follows:

$$f\left(\underset{x}{\rightarrow}\right) = \left(\frac{1}{6.931} - \frac{G_a G_b}{G_c G_d}\right)^2 \qquad (8)$$

where $\frac{1}{6.931}$ desired ratio, Ga, Gb, Gc, Gd teeth of gears A, B, C, and D respectively, with the ratio is:

$$\text{Gear Ratio} = \frac{G_a G_b}{G_c G_d} \qquad (9)$$

subject to: $\forall \{G_i ,\ \ 12 \le G_i \le 60\}$, where G_i is teeth of Ga, Gb, Gc, Gd.

Table 7 Comparative Result Gear train design problem

Algorithm	x_1 (G_a)	x_2(G_b)	x_3 (G_c)	x_4 (G_d)	Optimal Error	Ratio
SHOA	12	24	34.4	58.1	4.04E-15	0.1440
NL	18	22	45	60	5.70E-04	0.1466
CS	19	16	43	49	2.70E-12	0.1442
AZOA	60	17.52	12	24.29	0	3.606
MFO	43	19	16	49	2.70E-12	1.042

In Table 7 for SHOA with AZOA, MFO, Non-Linear (NL) [53], and Cuckoo Search (CS) [54] shown, the table presents gear teeth of A, B, C, and D, optimal, and ratio (x) as comparison parameters, where the ratio must be closer to (1/6.931). In this study, SHOA had high performance over other algorithms, first three algorithms shown in Table 8 had better performance than the last two because their ratio rates as constraints were also satisfied when compared with the AZOA and MFO.

B. THREE-BAR TRUSS DESIGN PROBLEM SOLVING

The three-bar truss problem is a civil engineering design problem whose objective is to achieve the minimum weight subjected to stress, deflection, and buckling constraints and evaluate the optimal cross-sectional area ($A1$, $A2$). Mathematically, to minimize the weight of a three-bar truss construction, according to [55], an objective function and constraints are formulated as follows:

$$\text{Minimize } f(x) = (2\sqrt{2}x_1 + x_2) \times l \tag{10}$$

Subject to:

$$C_1(x) = \frac{\sqrt{2}\,x_1 + x_2}{\sqrt{2}\,x^2{}_1 + 2x_1x_2} P - \sigma \le 0 \tag{11}$$

$$C_2(x) = \frac{x_2}{\sqrt{2}\,x^2{}_1 + 2x_1x_2} P - \sigma \le 0 \tag{12}$$

$$C_3(x) = \frac{1}{\sqrt{2}\,x_2 + x_1} P - \sigma \le 0 \tag{13}$$

$\forall i, 0 \le x_i \le 1, where\ i = 1{,}2$, the constant parameters are: $l = 100\ cm$, $P = 2\ KN/cm^2$, $\sigma = 2\ KN/cm^2$.

The comparative result shown in Table 8 presents the performance of the SHOA compared with AZOA, CS, MFO, and engineering design optimization by (Ray T, and Saini) TSa [55] algorithms, SHOA had tested nearly 20 times 30 rounds all minimum fitness was between (263.90, 263.91) and maximum fitness was between (264.0, 264.6), and the (mean, std) of 30 round is (264.026, 0.137), in the mentioned table shows the performance of SHOA either better or equals others with some ×10-4 points.

Table 8 Comparative Result Three-bar Truss Design Problem

Algorithm	x_2 [A_1]	x_2 [A_2]	Optimal weight
SHOA	0.7866268	0.4140837	263.9000
AZOA	0.7885471	0.408610	263.8958
TSa	0.795	0.395	264.3
MFO	0.7882447	0.4094669	263.8959
CS	0.788670	0.40902	263.9716

C. ANTENNA SPACED ARRAY PROBLEM SOLVING WITH SHOA

Optimization of antenna arrays means reducing the side-lobe level (SLL) of a non-uniformly spaced linear array. The fitness value for the problem has been formulated to the maximum SLL to optimize the non-uniformly spaced array [50,56]. Objective function and constraints are formulated as follows:

$$f\left(\underset{x}{\rightarrow}\right) = max[20\,log\,|G(\theta)|] \tag{14}$$

where the:

$$G(\theta) = \sum_{i=1}^{n} cos[2\pi x_i\,(cos\,\theta - cos\,\theta_s)] + cos[2.25 \times 2\pi(cos\,\theta - cos\,\theta_s)] \tag{15}$$

$$n = 4 \quad and\,\theta = 45° , \qquad \theta_b = 90°$$

Subject to:

$$d = |xi - xj| > 0.25\lambda \tag{16}$$

$$0.125\lambda < min\{x_i\} \le 2.0\lambda \tag{17}$$

$$x_i \in (0{,}2.25) ,\ i = 1{,}2{,}3{,}4.\ i \ne j$$

To minimize SLL, the element should optimize without violation of above constraints above, where $\theta\ is\ elevationangle, and\ \ \theta_b\ is\ beam\ angle$, xi which is an element of antenna must be greater than 0.125λ, the distance between elements must be more than 0.25λ.

The comparative assessment in Table 9 shows an optimal value between both SHOA and FDO algorithms, the SHOA found a minimum optimal out of 30 turns as shown in Table 9, and the maximum optimal value was (-169.995) with parameters (1.545, 0.203, 0.475, 1.254), the assessment shows the superiority of SHOA in all turns when compared with FDO.

Table 9 Comparative Result Antenna Spaced Array Problem

Algorithm	P_1	P_2	P_3	P_4	Optimal SLL
SHOA	1.345	1.346	1.346	1.346	-268.185
FDO	0.713	1.595	0.433	0.130	-120

D. FREQUENCY-MODULATED SOUND WAVE DESIGN PROBLEM SOLVING

The frequency modulation in sound waves required to find optimal parameters to transfer sounds, it has six parameters to

optimize as ($a1$, $w1$, $a2$, $w2$, $a3$, $w3$), which is a highly complex problem in the multimodal field, with fitness function is a minimum summation of square error between evaluated and modeled data, the fitness and constraints are formulated as follows:

$$f\left(\underset{p}{\rightarrow}\right) = \sum_{i=1}^{100}(y(t) - y_0(t))^2 \quad (18)$$

Where:

$$\left(\underset{p}{\rightarrow}\right) = (a1, w1, a2, w2, a3, w3) \quad (19)$$

$$y(t) = a_1.sin(w_1.t.\theta + a_2.sin(w_2.t.\theta + a_3.sin(w_3.t.\theta))) \quad (20)$$

$$y(t) = (1.0).sin((5.0).t.\theta + (1.5).sin((4.8).t.\theta + (2.0).sin((4.9).t.\theta))) \quad (21)$$

With $\theta = (2\pi/100)$, the range of the parameter is [-6.4, 6.35], and minimum fitness values are the optimal solution for the sound wave problems to transfer sound with the lowest error rate [57–59].

Table 10 Comparative result of frequency-modulated sound wave

Alg.	a_1	w_1	a_2	w_2	a_3	w_3	Error	avg.
SHOA	1.04	5.05	-1.36	-4.81	1.90	-4.87	1.062	6.3
FDO	0.97	-0.24	-4.31	-0.01	-0.57	4.93	3.220	NA
fGA	NA	NA	NA	NA	NA	NA	0.0	8.4

In Table 10 comparative results of a frequency-modulated wave for the current study result with FDO and Fork Genetic Algorithm (fGA) [57] are shown, the table presented six parameters, with the best fitness value out of 30 runs considered as an optimal result, and an average of 30 runs of the SHOA with the mentioned algorithms FDO and fGA, the unknown data is written as NA. The result shows that SHOA has a higher performance than both algorithms. The SHOA finds better fitness than FDO and a better average than fGA out of 30 runs. The FDO average result was NA, an optimal value has been generated depending on the presented parameters from FDO, and the fGA reached the optimal solution $\left(\underset{p}{\rightarrow}\right) = 0.0$, but the parameters had not been presented.

V. CONCLUSIONS

In this study, the theoretical oft for the novel SHO swarm-based algorithm has been provided via concepts of exploration and exploitation. It mimics the bird's breed adaptation and lifestyle in the population. The SHOA applied to 29 benchmark functions (single-modal, multi-modal, composite, and 100-Digit Challenge) test functions, and engineering problems (constrained, unconstrained) are solved. The performance has been compared with recent and powerful algorithms. The results demonstrated the effectiveness of the newly developed approach SHOA in solving all test functions and a variety of engineering problems and showed that this can provide reliable and accurate solutions in a variety of contexts. Through the SHOA study, the following were concluded:

- Faster convergence rate, the mechanism adapts the best birds for the next generation.
- More stable than compared algorithms, the balance of convergence and divergence leads to the best solution.
- Accurate search and high exploration and investigation promise a promising area of space within a reasonable amount of time.
- High performance in solving constrained and un-constrained multimodal real optimization problems.
- Highly multi-modal problem optimizer, because each nest is considered a population with an optimal solution, and all are considered a single population that finds the global optimum from local optimums.

The proposed SHOA is a single objective; for the future, many research works can be conducted in multi-objective, binary, and discrete versions, all of which can be used to solve a variety of types of problems.

ACKNOWLEDGMENT

The authors would like to sincerely thank the members of the software engineering department from the Slahaddin University-Erbil for their invaluable contributions that enabled this study to be completed successfully, special thanks to family members.

Appendix A

Table 11 Uni-modal Test Functions with dimension = 10

Function	Range	Shift	f_{min}
$F1(x) = \sum_{i=1}^{n} {x_i}^2$	[-100, 100]	[-30, -30, -30, ...]	0
$F2(x) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	[-10, 10]	[-3, -3, -3, …]	0
$F3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	[-100, 100]	[-30, -30, -30, ...]	0
$F4(x) = max_i\{\|x_i\|, 1 \leq i \leq n\}$	[-100, 100]	[-30, -30, -30, ...]	0
$F5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]	[-15, -15, -15, …]	0
$F6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	[-100, 100]	[-75, -75, -75, …]	0
$F7(x) = \sum_{i=1}^{n} i{x_i}^4 + rand[0,1]$	[-1.28, 1.28]	[-0.25, -0.25, -0.25, ...]	0

Table 12 Multi-modal Test Functions with dimension = 10

Function	*Range*	*Shift*	*min*
$F8(x) = \sum_{i=1}^{n} -{x_i}^2 \sin\left(\sqrt{\|x_i\|}\right)$	[-500, 500]	[-300, -300, -300, ...]	-418,9829
$F9(x) = \sum_{i=1}^{n} [{x_i}^2 - \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	[-2, -2, -2, ...]	0
$F10(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	[-32, 32]		0
$F11(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	[-400, -400, -400, ...]	0
$F12(x) = \frac{\pi}{n}\left\{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \; [1 + 10 \; \sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{(x+1)}{4}$ $u(x_i, a, k, m) = \left\{\begin{matrix} k(x_i - a)^m & x_i > a \\ 0 - a < & x_i < a \\ k(-x_i - a)^m & x_i < -a \end{matrix}\right\}$	[-50, 50]	[-30, -30, -30, …]	0

$F13(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2[1+sin^2(3\pi x_i+1)] + (x_n-1)\,[1+sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n}u(x_i,5,100,4)$ $u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 - a < x_i < a & \\ k(-x_i-a)^m & x_i < -a \end{cases}$	[-50, 50]	[-10, -10, -10, …]	0

Table 13 Composite Test Functions with dimension = 10, Range [-5,5], $f_{min} = 0$

Functions
F14 (CF1) $f1, f2, \dots\dots, f10 = \ sphere\ function$ $[\delta_1, \delta_2, \delta_3 \dots \delta_{10}] = [1,1,1,\dots,1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[\frac{5}{100}, \frac{5}{100}, \dots, \frac{5}{100}\right]$
F15 (CF2) $f1, f2, \dots\dots, f10 = Griewank's\ function$ $[\delta_1, \delta_2, \delta_3 \dots \delta_{10}] = [1,1,1,\dots,1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[\frac{5}{100}, \frac{5}{100}, \dots, \frac{5}{100}\right]$
F16 (CF3) $f1, f2, \dots\dots, f10 = Griewank's\ function$ $[\delta_1, \delta_2, \delta_3 \dots \delta_{10}] = [1,1,1,\dots,1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [1,1,\dots,1]$
F17 (CF4) $f1, f2 = Ackley's function$ $f3, f4 = \ Rastrigin's\ function$ $f5, f6 = \ Weierdtrass\ function$ $f7, f8 = Griewank's\ function$ $f9, f10 = \ Sphere\ function$ $[\delta_1, \delta_2, \delta_3 \dots \delta_{10}] = [1,1,1,\dots,1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[\frac{5}{32}, \frac{5}{32}, 1, 1, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100}\right]$
F18 (CF4) $f1, f2 = Rastrigin's\ function$ $f3, f4 = \ Weierdtrass\ function$ $f5, f6 = Griewank's\ function$ $f7, f8 = Ackley's function$ $f9, f10 = \ Sphere\ function$ $[\delta_1, \delta_2, \delta_3 \dots \delta_{10}] = [1,1,1,\dots,1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[\frac{1}{5}, \frac{1}{5}, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{32}, \frac{5}{32}, \frac{5}{100}, \frac{5}{100}\right]$
F18 (CF4) $f1, f2 = Rastrigin's\ function$ $f3, f4 = \ Weierdtrass\ function$ $f5, f6 = Griewank's\ function$ $f7, f8 = Ackley's function$ $f9, f10 = \ Sphere\ function$ $[\delta_1, \delta_2, \delta_3 \dots \delta_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1]$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[0.1 \times \frac{1}{5}, \ 0.2 \times \frac{1}{5}, \ 0.3 \times \frac{5}{0.5}, 0.4 \times \frac{5}{0.5}, \ 0.5 \times \frac{5}{100}, \ 0.6 \times \frac{5}{100}, \ 0.7 \times \frac{5}{32}, 0.8 \times \frac{5}{32}, 0.9 \times \frac{5}{100}, 1 \times \frac{5}{100}\right]$

Table 14 Summary of basic "The Hundred-Digit Challenge" Benchmarks

Name	Functions	Dim	Range
C01	STRONG CHEBYSHEV POLYNOMIAL FITTING PROBLEM	9	[-8192,8192]
C02	INVERSE HILBERT MATRIX PROBLEM	16	[-16384, 16384]
C03	LENNARD-JONES MINIMUM ENERGY CLUSTER	18	[-4,4]
C04	RASTRIGIN'S FUNCTION	10	[-100, 100]
C05	GRIEWANGK'S FUNCTION	10	[-100, 100]
C06	WEIERSTRASS FUNCTION	10	[-100, 100]
C0	MODIFIED SCHWEFEL'S FUNCTION	10	[-100, 100]
C08	EXPANDED SCHAFFER'S F6 FUNCTION	10	[-100, 100]
C09	HAPPY CAT FUNCTION	10	[-100, 100]
C10	ACKLEY FUNCTION	10	[-100, 100]

APPENDIX B

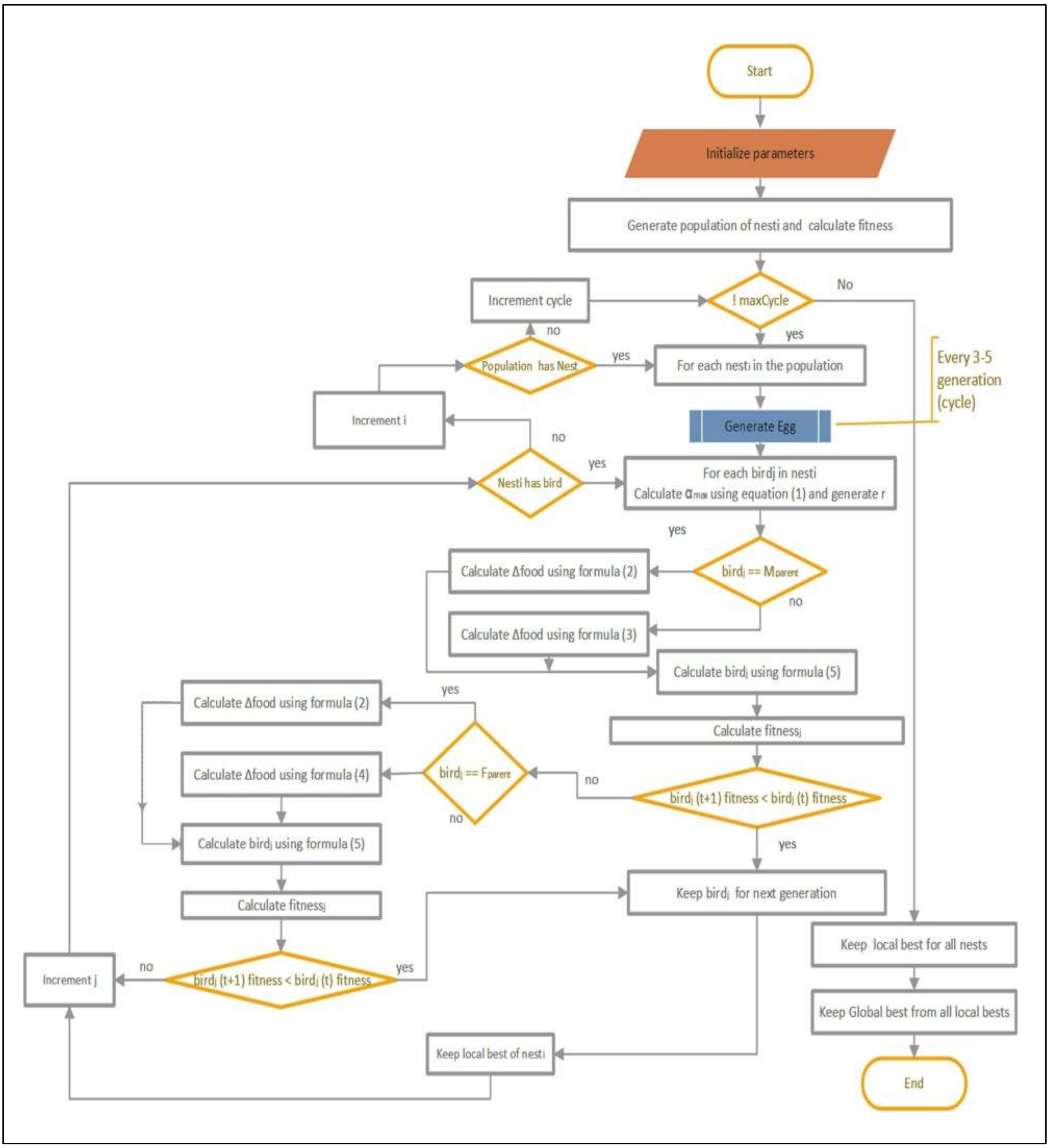

Figure 6 Flow chart of SHOA

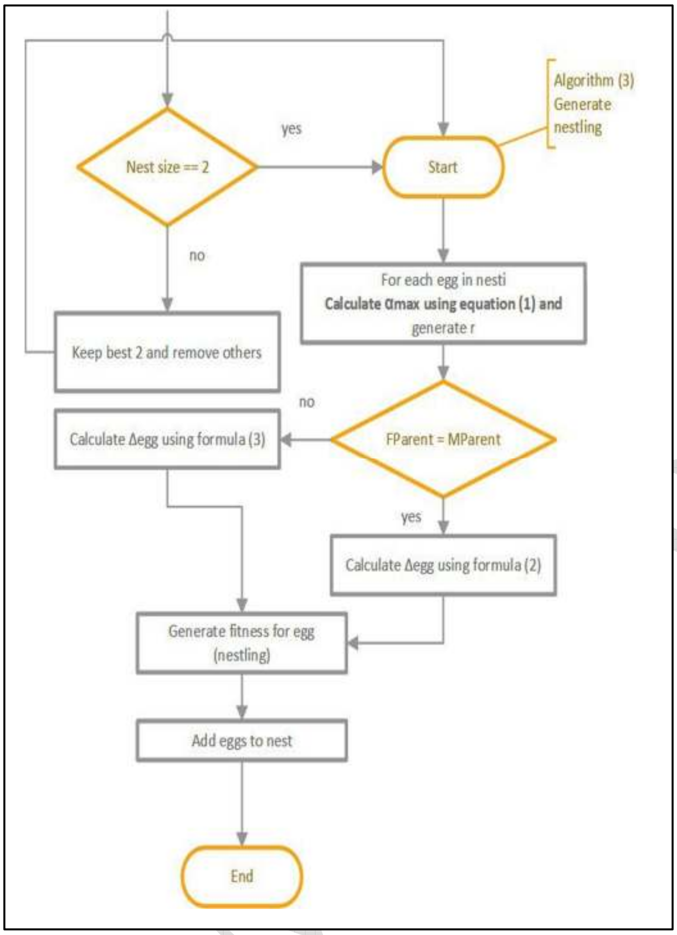

Figure 7 Flow chart of Generate Egg

REFERENCES

[1] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. Science (80-) 1983;220:4598. https://doi.org/http://science.sciencemag.org/.
[2] Greiner R. PALO: A probabilistic hill-climbing algorithm. Artif Intell 1996;84:177–208. https://doi.org/10.1016/0004-3702(95)00040-2.
[3] Hansen P, Mladenović N. Variable neighborhood search. Handb Heuristics 1997;1–2:1097–100. https://doi.org/10.1007/978-3-319-07124-4_19.
[4] Glover F. Tabu Search: A Tutorial. Interfaces (Providence) 1990;20:74–94. https://doi.org/10.1287/inte.20.4.74.
[5] Holland JH 1975. Adaptation in Natural and Artificial Systems:An introductory analysis with applications to biology, control, and artificial intelligence. 2nd Editio. The MIT Press; n.d. https://doi.org/https://psycnet.apa.org/record/1975-26618-000.
[6] RAINER Storn KP. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Australas Plant Pathol 1997;38:284–7. https://doi.org/10.1071/AP09004.
[7] Koza. JR. Hierarchical Genetic Algorithms Operating on Populations of Computer Programs. Hierarchical Genet Algorithms Oper Popul Comput Programs 1989;1:768–774.
[8] Dorigo, M., Birattari, M., & Stutzle T. Ant Colony Optimization. IEEE Comput Intell Mag 2006;1(4):28–39. https://doi.org/doi:10.1109/mci.2006.329691.
[9] Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperating agents. IEEE Trans Syst Man, Cybern Part B Cybern 1996;26:29–41. https://doi.org/10.1109/3477.484436.
[10] Dorigo M. Learning and Natural Algorithms 1992.
[11] Martens D, De Backer M, Haesen R, Vanthienen J, Snoeck M, Baesens B. Classification with ant colony optimization. IEEE Trans Evol Comput 2007;11:651–65. https://doi.org/10.1109/TEVC.2006.890229.
[12] Zuo L, Shu L, Dong S, Zhu C, Hara T. A multi-objective optimization scheduling method based on the ant colony

algorithm in cloud computing. IEEE Access 2015;3:2687–99. https://doi.org/10.1109/ACCESS.2015.2508940.
[13] Eberhart R, Kennedy J. New optimizer using particle swarm theory. Proc Int Symp Micro Mach Hum Sci 1995:39–43. https://doi.org/10.1109/mhs.1995.494215.
[14] Braik M, Sheta A, Ayesh A. Image Enhancement Using Particle Swarm Optimization. Int J Innov Comput Appl 2007;1:138–45.
[15] Mikki SM, Kishk AA. Quantum particle swarm optimization for electromagnetics. IEEE Trans Antennas Propag 2006;54:2764–75. https://doi.org/10.1109/TAP.2006.882165.
[16] Ray T, Liew KM. Society and civilization: An optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 2003;7:386–96. https://doi.org/10.1109/TEVC.2003.814902.
[17] Karaboga D. An idea based on honey bee swarm for numerical optimization. Tech ReportTR06, Erciyes Univ Eng Fac Comput Eng Dep 2005.
[18] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. J Glob Optim 2007;39:459–71. https://doi.org/10.1007/s10898-007-9149-x.
[19] Gao W, Liu S, Huang L. A global best artificial bee colony algorithm for global optimization. J Comput Appl Math 2012;236:2741–53. https://doi.org/10.1016/j.cam.2012.01.013.
[20] Uzer MS, Yilmaz N, Inan O. Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification. Sci World J 2013;2013. https://doi.org/10.1155/2013/419187.
[21] Bullinaria JA, AlYahya K. Artificial Bee Colony training of neural networks: Comparison with back-propagation. Memetic Comput 2014;6:171–82. https://doi.org/10.1007/s12293-014-0137-7.
[22] Yao B, Yan Q, Zhang M, Yang Y. Improved artificial bee colony algorithm for vehicle routing problem with time windows. PLoS One 2017;12:1–18. https://doi.org/10.1371/journal.pone.0181275.
[23] El-abd M. Generalized Opposition-Based Artificial Bee Colony Algorithm 2012.
[24] Passino KM. Bacterial Foraging Optimization. Int J Swarm Intell Res 2010;1:1–16. https://doi.org/10.4018/jsir.2010010101.
[25] Patnaik SS, Panda AK. Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter. Appl Comput Intell Soft Comput 2012;2012:1–10. https://doi.org/10.1155/2012/897127.
[26] Muñoz MA, Halgamuge SK, Alfonso W, Caicedo EF. Simplifying the bacteria foraging optimization algorithm. 2010 IEEE World Congr Comput Intell WCCI 2010 - 2010 IEEE Congr Evol Comput CEC 2010 2010:18–23. https://doi.org/10.1109/CEC.2010.5586025.
[27] Yang XS. Firefly algorithms for multimodal optimization. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2009;5792 LNCS:169–78. https://doi.org/10.1007/978-3-642-04944-6_14.
[28] Gebremedhen HS, Woldemichael DE, Hashim FM. A firefly algorithm based hybrid method for structural topology optimization. Adv Model Simul Eng Sci 2020;7. https://doi.org/10.1186/s40323-020-00183-0.
[29] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Syst 2015;89:228–49. https://doi.org/10.1016/j.knosys.2015.07.006.
[30] Pedram Haeri Boroujeni S, Pashaei E. Data Clustering Using Moth-F lame Optimization Algorithm. ICCKE 2021 - 11th Int Conf Comput Eng Knowl 2021:296–301. https://doi.org/10.1109/ICCKE54056.2021.9721483.
[31] Wan Afandie WNEA, Rahman TKA, Zakaria Z. Optimal load shedding using Bacterial Foraging Optimization Algorithm. Proc - 2013 IEEE 4th Control Syst Grad Res Colloquium, ICSGRC 2013 2013:93–7. https://doi.org/10.1109/ICSGRC.2013.6653282.
[32] Mangaraj BB, Jena MR, Mohanty SK. Bacteria Foraging Algorithm in Antenna Design. Appl Comput Intell Soft Comput 2016;2016. https://doi.org/10.1155/2016/5983469.
[33] Skackauskas J, Kalganova T, Dear I, Janakiram M. Dynamic impact for ant colony optimization algorithm. Swarm Evol Comput 2022;69. https://doi.org/10.1016/j.swevo.2021.100993.
[34] Dokeroglu T, Sevinc E, Cosar A. Artificial bee colony optimization for the quadratic assignment problem. Appl Soft Comput J 2019;76:595–606. https://doi.org/10.1016/j.asoc.2019.01.001.
[35] Z. Xu CF. Design of Available Land Space Optimization Model Based on Ant Colony Optimization Algorithm. IEEE Int. Conf. Integr. Circuits Commun. Syst., Raichur, India: 2023, bl 1–6. https://doi.org/10.1109/ICICACS57338.2023.10099535.
[36] Qolomany B, Ahmad K, Al-Fuqaha A, Qadir J. Particle Swarm Optimized Federated Learning for Industrial IoT and Smart City Services. 2020 IEEE Glob Commun Conf GLOBECOM 2020 - Proc 2020. https://doi.org/10.1109/GLOBECOM42002.2020.9322464.
[37] Zou R, Kalivarapu V, Winer E, Oliver J, Bhattacharya S. Particle Swarm Optimization-Based Source Seeking. IEEE Trans Autom Sci Eng 2015;12:865–75. https://doi.org/10.1109/TASE.2015.2441746.
[38] Djemai T, Stolf P, Monteil T, Pierson JM. A discrete particle swarm optimization approach for energy-efficient IoT services placement over fog infrastructures. Proc - 2019 18th Int Symp Parallel Distrib Comput ISPDC 2019

2019;2019:32–40. https://doi.org/10.1109/ISPDC.2019.00020.
[39] Abdulkarim HK, Rashid TA. Moth-Flame Optimization and Ant Nesting Algorithm : A Systematic Evaluation. Proc. 1st Int. Conf. Innov. Inf. Technol. Bus. (ICIITB 2022), vol. 1, Online: Atlantis Press; 2023, bl 139–52. https://doi.org/10.2991/978-94-6463-110-4_11.
[40] Hill C, Miles K, Maddox K, Tegeler A. y to aid mate acquisition?Are adoptions in a predatory songbird a strateg. J F Ornithol 2023;94:0–5. https://doi.org/10.5751/jfo-00284-940209.
[41] Yosef R, Grubb TC. Territory Size Influences Nutritional Condition in Nonbreeding Loggerhead Shrikes (Lanius ludovicianus): A Ptilochronology Approach . Conserv Biol 1992;6:447–9. https://doi.org/10.1046/j.1523-1739.1992.06030447.x.
[42] Gawlik DE, Bildstein KL. Reproductive success and nesting habitat of Loggerhead Shrikes in north-central South Carolina. Wilson Bull 1990;102:37–48. https://doi.org/http://www.ncbi.nlm.nih.gov/pubmed/24671971.
[43] Ridgway R. A manual of north american birds. PHILADEPHIA: J.B. LIPPNCOTT COMPANY: Contributed by University of Toronto - Gerstein Science Information Centre.; 1887. https://doi.org/https://www.biodiversitylibrary.org/page/21816472.
[44] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evol Comput 1999;3:82–102. https://doi.org/10.1109/4235.771163.
[45] Liang JJ, Suganthan PN, Deb K. Novel composition test functions for numerical global optimization. Proc - 2005 IEEE Swarm Intell Symp SIS 2005 2005;2005:71–8. https://doi.org/10.1109/SIS.2005.1501604.
[46] Marcin Molga CS. Test Functions for optimization algorithm 2005:205–9.
[47] Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 2011;1:3–18. https://doi.org/10.1016/j.swevo.2011.02.002.
[48] Iverson BL, Dervan PB. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization n.d.:7823–30.
[49] K. V. Price, N. H. Awad, M. Z. Ali PNS. Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization. Tech Rep Nanyang Technol Univ Singapore 2018:22.
[50] Abdullah JM, Ahmed T. Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process. IEEE Access 2019;7:43473–86. https://doi.org/10.1109/ACCESS.2019.2907012.
[51] Mohapatra S, Mohapatra P. American zebra optimization algorithm for global optimization problems. vol. 13. Nature Publishing Group UK; 2023. https://doi.org/10.1038/s41598-023-31876-2.
[52] Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 2002;191:1245–87. https://doi.org/https://doi.org/10.1016/S0045-7825(01)00323-1.
[53] Sandgren E. Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des Trans ASME 1990;112:223–9. https://doi.org/10.1115/1.2912596.
[54] Gandomi AH, Yang XS, Alavi AH. Erratum: Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems (Engineering with Computers DOI:10.1007/s00366-011-0241-y). Eng Comput 2013;29:245. https://doi.org/10.1007/s00366-012-0308-4.
[55] Ray T, Saini P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 2001;33:735–48. https://doi.org/10.1080/03052150108940941.
[56] Jin N, Rahmat-Samii Y. Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations. IEEE Trans Antennas Propag 2007;55:556–67. https://doi.org/10.1109/TAP.2007.891552.
[57] Tsutsui S, Fujimoto Y. Forking Genetic Algorithm with Blocking and Shrinking Modes (fGA). Proc 5th Int Conf Genet Algorithms, ICGA '93 1993:206–15.
[58] Herrera F, Lozano M. Gradual distributed real-coded genetic algorithms. IEEE Trans Evol Comput 2000;4:43–62. https://doi.org/10.1109/4235.843494.
[59] Das S, Suganthan PN. Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems. Electronics 2011:1–42.

Hanan K. AbdulKarim received B.Sc. (Hons.) and M.Sc. degrees in Software Engineering from Salahaddin University-Erbil Iraq, in 2008 and 2014 respectively. She is currently pursuing a Ph.D. in Software Engineering from Salahaddin University, College of Engineering, Erbil, Iraq in swarm intelligence. She worked as a teaching assistant for two years in 2009, then started her M.Sc. study, and worked as an IT manager and instructor in the same university in 2014, where currently studying.

TARIK A. RASHID received a Ph.D. degree in computer science and informatics from the College of Engineering, Mathematical and Physical Sciences, University College Dublin (UCD), Ireland in 2006. He joined the University of Kurdistan Hewlêr, in 2017. He is a Principal Fellow for the Higher Education Authority (PFHEA-UK) and a professor in the Department of Computer Science and Engineering at the University of Kurdistan Hewlêr (UKH), Iraq. He is on the prestigious Stanford University list of the World's Top 2% of Scientists for the years 2021, 2022, and 2023. His areas of research cover the fields of Artificial Intelligence, Nature Inspired Algorithms, Swarm Intelligence, Computational Intelligence, Machine Learning, and Data Mining.